Interface Quotas and Internet-Derived Value

Bob Colwell

For the past few years, I've been pondering two theories: the Theory of Interface Quotas and the Theory of Required Internet-Derived Value. I'm working on more general theories for both, but that may take a few more years.

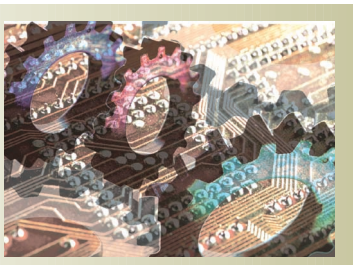
INTERFACE QUOTAS

My interface quota theory is entirely self-serving, and it probably stems from my desire to stop feeling like the old dog staring resentfully at the teacher of new tricks. According to this theory, we're all born with an innate quota of machine interfaces that we can become comfortable with.

In your lifetime, you'll learn hundreds of machine or system interfaces, but, as with languages, you'll be fluent only in those you learn when you're young. The rest you'll "speak" with an accent, or not at all. Eventually, any additional new interfaces will be inadvertently mapped to pieces of those already learned, which will bring you great trouble, and you'll give up on the new machine in disgust.

In other words, youngsters eagerly internalize new technology, but eventually we reach a limit, and in that instant it becomes much more appealing to criticize the new interface than to learn it.

For example, you've learned how to use TVs, clock radios, watches, computers, automobiles, DVD players, iPods, cameras, cell phones, tools, printers, ATMs, airline kiosks, and a host of other devices. It's a list of length *N*, for *N* a large number.



Eventually, we reach a limit where it's more appealing to criticize a new interface than to learn it.

But only a few days after getting item *N* working, your cell phone's keypad no longer dials the number 7, there's a dark band across the top of the LCD screen, the batteries only last about 15 minutes on a charge, and your ring tone is so dated that mothers cover the ears of their children when they hear it and hustle them away from you. You bow to the inevitable and get a new cell phone, with the salesperson's infinite list of wonderful new features ricocheting around in your head like a box of ping-pong balls dropped onto a wood floor.

In the past, you would have taken that cell phone home and put it through its paces, programming its address list, configuring it for Web access and instant messaging, setting its phaser to stun. But shockingly, this time you find you no longer care. You bought it to be a phone, and that's all you want from it. *The War and Peace*-sized manual goes into the box, and the box goes into recycling. You've reached your interface limit: The part of your cranium that handles obtuse and completely artificial machine interfaces is full, and there's no more room for yet another one. Your machine interface neurons have linked their metaphorical elbows and formed a skirmish line: None shall pass.

At first, you think this is only temporary. Maybe it's related to the curry overdose from the Thai restaurant last night. Or maybe you're a Cleveland Browns fan, and a huge fraction of your brain is tied up in trying to remember why.

Then your 17-year-old son walks in and says, "Wow, Dad, what a cool new phone. I heard about this model. It has a 3D camera with X-ray vision, mental telepathy input, satellite Web access, GPS, faster-than-light instant messaging, and a ring of awesome little lights that makes it look as though it's levitating in the dark. It can control your blender and make margaritas. What have you programmed it to do so far?" He sees your blank stare, and his eyes go wide with dismay. "Dad, you haven't programmed it at all, have you? You still have the default ring tones, don't you?"

Your son slowly backs away from you; you can see in his eyes that he is reevaluating his entire relationship with you. You can no longer be the cool guy he mistook you for when he was 10 years old. Your connection with him has irrevocably changed. He

Published by the IEEE Computer Society

doesn't see the problem. He still thinks it's great fun to explore the latest techno-gadget and tease out its incredible facilities.

But there's nothing you can do. You simply can't learn another interface, no matter the cost in broken human relationships. It's sad, so sad.

MISSPENT YOUTH

When I was an undergrad, anyone using computers learned that poring over manuals, looking for command-line switches and hidden utilities, was time well spent. You could "nice" your jobs to higher priority, give yourself larger swap partitions, hide large files, and sling mag tapes like Rambo's bandoliers. Your friends came to regard you as a techno-wizard. Little did you know then, however, that you were pumping your cranium full of bits that were going to be useful only for a short time; 30 years later, those bits would turn to obstructionary sludge, indirectly causing a rift with your son.

Fortunately, there's a reasonably effective gambit to use in the cell phone saga. You hand your new phone to your son and say, "I'll pay you $25 to set this phone up the way you would use it." An hour later, he hands it back to you, and, apart from the fact that your quick dial list now identifies your son's listing as "Darling Son" and your new ring tone is some kind of revolting gurgling to a hip-hop beat, it's money well spent.

In some ways, we're doing much better at human-machine interfaces. When I was a child, if you wanted to watch television, you had to understand horizontal hold, vertical hold, fine tuning, and which way to orient the rabbit-ear antenna for the particular station you wanted to watch. Since there were only three stations, this wasn't as hard as it would be today. However, the aluminum foil kept falling off the rabbit ears, and, depending on where the TV viewers were sitting, the picture would phase in and out. When someone entered the room, it was common for viewers to snap, "Sit down and don't move!" so the picture would stabilize.

Another example is the engine control of automobiles. In the 1960s, many cars had a choke control on the dashboard. To start the engine, you had to pull the choke out; as the engine warmed up, you pushed the choke back in to keep the engine from stalling.

Digital cameras have taken up the slack in introducing interface complexity.

I still wonder how many people had any real idea of what that control did, why it was necessary then, and for that matter why it's not necessary anymore. Hint: Your engine computer is your friend—except in the 1984 Jeep Cherokee I had, in which the computer's ground turned out to be intermittent. Computers object to that kind of treatment, and take their petty revenge by removing all vehicle propulsion at the most inopportune times.

I used to cite VCRs as examples of impossible user interfaces, but nobody uses them anymore. TiVo seems to have wiped them out—another example of adding complexity in the right place to make the user interface simpler. If my wristwatch had a way to learn from, say, the infrared port of my laptop, it might actually know the month and date. As it is, those have been random numbers for several years now, another victim of the Theory of Interface Quotas.

But where TiVo has fixed VCRs, digital cameras have taken up the slack in introducing interface complexity. If all you want to do is point and shoot, these cameras do a good job—assuming you can figure out how to download the pictures and reformat the flash memory. If you want to do more than that, though, you should first admit that you're in big trouble, and pray for guidance.

NO THINKING

Steve Krug has written a delightful book titled *Don't Make Me Think: A Common Sense Approach to Web Usability* (New Riders, 2000). In his winning writing style, Krug addresses the topic of creating a great Web site from every angle: navigation, organization, fonts and sizes, graphics, icons, writing, colors, and speed at which the site loads.

Krug also relates an eerily familiar story. It seems that in *A Study in Scarlet*, Dr. Watson discovers that Sherlock Holmes doesn't know that the Earth goes around the sun. Holmes replies that the finite capacity of the human brain dictates that he not have "useless facts elbowing out the useful ones." Precisely, my dear Watson.

One of Krug's fundamental insights is that if visitors have to think about how to use a Web site, that site has already failed in some important ways. After all, they didn't come to the site hoping for an interesting logic puzzle or an intellectual maze. They came with something else in mind entirely, and any cognitive effort that the site itself requires will be a distraction or hindrance.

Using real Web sites, Krug offers numerous examples of good sites and bad sites. He also shows several years of evolution of Web sites such as Amazon's, to graphically illustrate the learnings such major Web presences have collected.

If Krug has his way, I won't have to get my son to navigate for me around the Web because the interfaces will have been engineered out of the way. Horizontal hold will have been absorbed into the site's innards, leaving me free to enjoy the show. No additional cranial strain.

Software safety expert Nancy Leveson says that when replacing old mechanical system controls of, say, a chemical plant, the new computer-based controls should leverage what

the plant operators already know (*Safeware*, Addison-Wesley, 1995). "Because humans often revert to stereotype, stereotypes should be followed in design. This criterion includes making computer displays look similar to the analog displays they are replacing." Aircraft pilots expect throttles to be in a certain place and to behave a certain way. When designing a fly-by-wire plane, it's a really good idea not to put levers where the throttles used to be or to have them control something other than engine speed.

Charles Perrow's analysis of the accident at Three Mile Island indicates that operator confusion was a substantial part of the problem (*Normal Accidents*, Princeton Press, 1999). Evidently, it was fairly common for some of the hundreds of valves and myriad complex subsystems to be set wrong, despite locks and logs expressly aimed at preventing such a misconfiguration of the reactor.

When the operators at TMI began a routine test on an accidentally misconfigured reactor, they unwittingly initiated a series of events that culminated in a near-Chernobyl-style meltdown. At many junctures in a drama extending over several days, reactor operators misunderstood apparently conflicting pressure gauges and made the problem worse. In effect, they had to infer the state of certain first-order aspects of the reactor, most importantly the level of the cooling water covering the reactor core. Although it's not clear why, their interface to the system purposely had no direct measurements available.

Designing safe, effective human interfaces is difficult under the best of circumstances, but it can be a matter of life or death in such large plant designs.

ME AND MY BROWSER

Here's what I want: Every electronic device and computer-based widget should communicate civilly with my computer. My computer can roam the Internet far and wide seeking whatever information is needed to tame, say, my watch or my guitar amplifier. But when I interact with those devices, I want to be looking at my familiar Windows interface.

Designing safe, effective human interfaces can be a matter of life or death.

To be clear: I make no claims that Windows is the best interface out there. I make no claims that Microsoft invented it. I don't care about those things. I only know that I have learned that interface, and I know it's general enough that it can be used for a great many other interfaces that I otherwise won't learn.

I don't want to fall off the technobandwagon just because I maxed out my interface capacity in my reckless youth. I just want to sit at my computer and set my watch, check my car's engine performance and service record, program my guitar amplifier, program the burglar alarm system, and monitor the heating system in my house, all from my browser window.

The observation that electronic devices are no longer independent stand-alone products led to my Theory of Required Internet-Derived Value. According to this theory, all electronic devices must connect to the Internet, and they must derive value from doing it, or they'll be supplanted by devices that can and do.

When I park my car in my garage at night, I want the engine computer to wirelessly talk to my PC, uploading the latest information on the car's status. Then I envision my PC traversing the Internet, comparing my car's age, model, and performance to that of equivalent cars to get an early warning of any impending problems. The environmental quality inspector can use this same radio connection instead of having to crawl under my dashboard to find the connector. Likewise, toll plazas could be negotiated directly by the car and the plaza, without requiring driver intervention.

When all devices can be routinely expected to talk to PCs and the Internet, their owners will be able to more readily maintain, repair, and program them. Do you have an impressive pile of remote control units littering your family room, as I do? When I tell my PC what electronics I have in the room, it should be able to fetch the relevant infrared control codes from the Internet and then automatically program one universal remote control.

Make things simpler. When designing, reuse interfaces we already know. The neurons you save may be your own. ■

Bob Colwell *was Intel's chief IA32 architect through the Pentium II, III, and 4 microprocessors. He is now an independent consultant. Contact him at bob.colwell@comcast.net.*